

SACM

Internet-Draft

Intended status: Informational

Expires: May 4, 2017

J. Schaad

August Cellars

D. Waltermire

National Institute of Standards and Technology

October 31, 2016

Prospective Architecture for SACM
draft-schaad-sacm-architecture-00

Abstract

This document describes the high level architecture for Security Automation and Continuous Monitoring (SACM). The architecture identifies the components that provide for the collection, storage, dissemination, and evaluation of posture information. This architecture also describes the interfaces and associated operations that define the interactions between these components. This information will inform future engineering work around identifying existing standards for collecting, storing, disseminating, and evaluating endpoint posture information. This architecture will also help in identifying standardization gaps that require new engineering effort.

Security practitioners need to request, analyze, and aggregate posture information from disparate sources that use differing means to identify endpoints, hardware, software, and configurations. This task is made harder by the large number of different protocols and formats needed to bring together all of this information into a single view. This architecture provides a means to automatically gather posture data together for standardized dissemination to downstream components. This allows security practitioners that leverage this architecture to focus on managing security problems, not data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

Internet-Draft

Prospective Architecture for SACM

October 2016

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 4, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	SACM Components	4
2.1.	Combining Roles	7
2.2.	SACM Collection	8
2.2.1.	Use of Existing Management Protocols	10
2.2.2.	The Need for Collection Choreography	10
2.2.3.	Collected Information Dissemination	12
3.	SACM Protocols	12
4.	Information Model	14
5.	Data Model	15
6.	IANA Considerations	15
7.	Security Considerations	15
8.	Informational References	15
	Acknowledgments	16
	Authors' Addresses	17

[1.](#) Introduction

This document represents some views of the authors but should not be considered to be the opinions of them either individually or collectively. The document is designed both to clarify the thinking

of the authors as well as to create some discussion on what the architecture looks like. As such, it is entirely reasonable that section of the document (i.e. the one on the IM) should be combined, removed or replaced. This can be either because they do not belong in this document or because they are just wrong.

Another aim of the document is to clarify what pieces of the work may need to be done in the future as this may help clarify some of the discussions are underway or have been completed. This is the aim of the list of protocols associated with a component. With this, we can perhaps start looking at what DM(s) are doing when they are sending data.

Jim: Part of the questions that would need to be addressed before publication is the audience of the document. It is my personal feeling that the document should be sufficiently complete that it can be handed to a person who is not in the community, and they should be able to understand all of the pieces and how they interact.

To manage information system security risk, network operators must know the operational state of the endpoints they manage, and must be able to assess known vulnerabilities against this operational state to understand potential security risks. This helps the network operator to determine which threats a given endpoint might be subject to, and supports decision making around actions that mitigate the associated risk. Vulnerability assessment is an example of this type of analysis, where endpoint software inventory is compared with known vulnerable software versions reported by software vendors and vulnerability researchers to determine which endpoints have vulnerable software. Knowledge of vulnerable software on endpoints can then drive software patching activities, reducing the attack surface of the network. Additionally, many organizations develop compliance requirements based on their understanding of security risks. These compliance requirements, also called policies, define what software may be installed on a given endpoint and how that software must be configured to achieve a sufficient level of security risk reduction for the organization. In both cases, an organization needs to have an up-to-date view of the operational state of the endpoints connected to their networks and accessing their information systems.

Understanding the operational state of an endpoint (hereafter

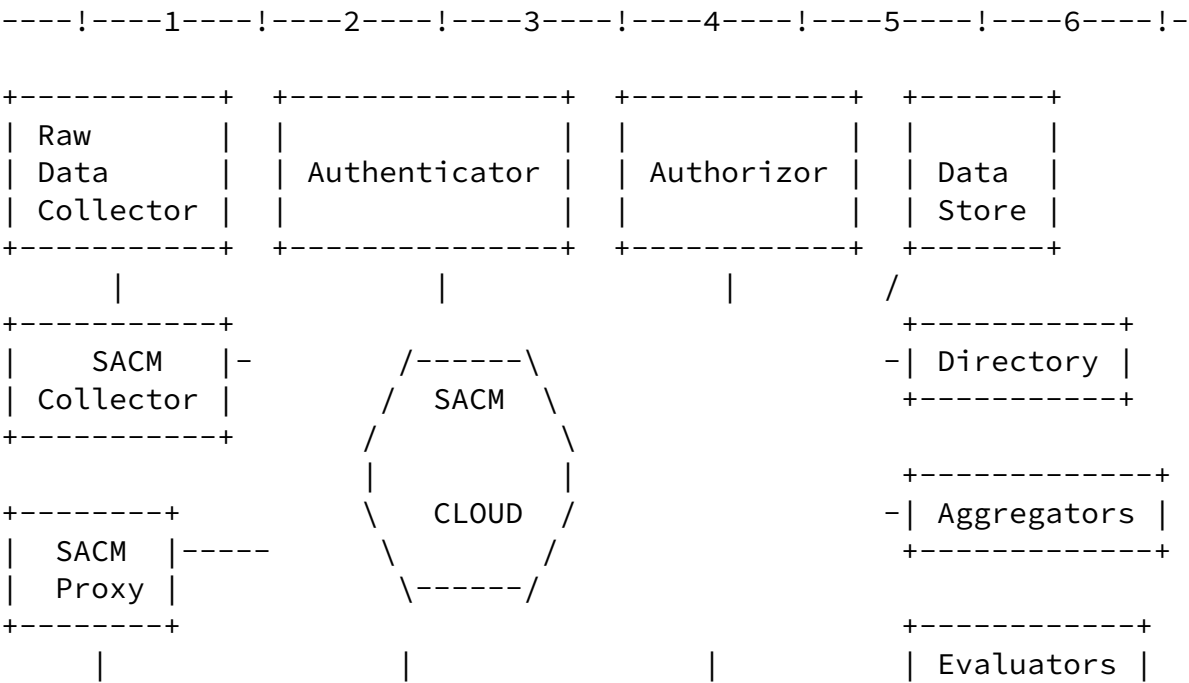
referred to as "endpoint posture") requires the ability to collect and evaluate endpoint posture information as posture changes occur. In order to make decisions quickly to prevent, deflect, or mitigate an attack, network operators must be able to collect, disseminate, and evaluate posture information within narrow timeframes. Making decisions quickly relies on automating posture collection and evaluation processes so that network operators can identify endpoints, software, and software configurations in a meaningful and enduring way that is suitable for trending and longer-term analysis. This document describes a suitable architecture that supports automated collection, storage, dissemination, and evaluation of endpoint posture information.

Still needs text dealing with other uses.

There is an IM and a DM. Some differences between these are found in [\[RFC3444\]](#).

2. SACM Components

The SACM architecture is made up of a number of different entities each of which supports one or more roles. A picture of what roles can exist can be found in Figure 1.



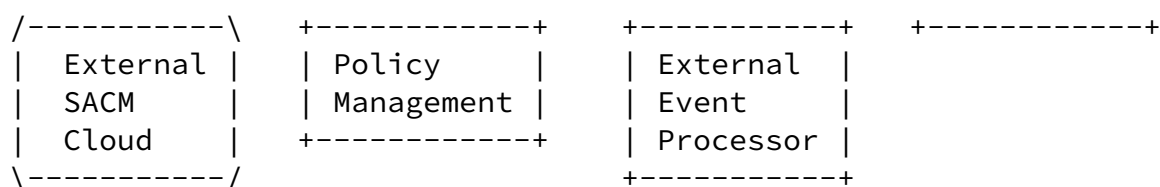


Figure 1: SACM Architecture

A description of the different roles in the picture above is as follows:

Authenticator: is a role that provides authentication services for other entities within the architecture. Entities can interact with an authenticator once and be provided with a long term credential, such as a certificate, or can be interacted with whenever authentication is needed, such as an EAP server. Even when interacting on a frequent basis, a short term credential such as a SAML assertion can be provided. Interactions with an authenticator include:

- * **Enrollment:** The act of being introduced to the authentication system.
- * **Revocation:** The act of being removed from the authentication system.
- * **Authentication:** The act of proving an identity to the authentication system matches one that was introduced to it.

Authorizer: is a role that provides authorization and access control to data and capabilities for an entity within the architecture. Frequently, but not always authorizers and authenticators will be co-located and administered. Interactions with an authorizer include:

- * **Authorization:** An administrative act of adding authorizations for an identity.
- * **Revocation:** An administrative act of removing authorizations for an identity.
- * **Querying:** The act of asking if an identity is authorized for

performing an action or receiving data.

Directory: is a role that provides methods to locate where services and data can be found in the architecture. Interactions with a directory include:

- * Publishing: The act of telling a directory what services or data an entity provides.
- * Unpublishing: The act of removing from a directory a list of services or data that an entity provides.
- * Querying: The act of asking a directory where a service or set of data can be found.
- * Directory Location: The act of finding a directory in the first place.

Data Store: is a role that provides a repository for data to be published into and queried from. Interactions with a data store include:

- * Publishing: The act of providing data to a data store.
- * Querying: The act of obtaining data from a data store.

- * Synchronization: The act of two data stores making the data they hold consistent.

SACM Collector: is a role that obtains data and then publishes it into the SACM environment. A SACM collector will either publish the data to a data store, or act as a data store itself. SACM collectors can location and publish data either in response to a query, if they act as a data store, on a schedule, or in response to some event. A SACM collector will format data provided to it into the SACM Information model, decorating it as necessary, and then provide it to the rest of the environment.

Raw Data Collector: is a role that is ancillary to the SACM architecture rather than being a core component. Raw data collectors gather data which is fed to a SACM collector. Examples

of raw data collectors could be NEA components, asset management databases (hardware, software or wetware), or network state databases such as a DHCP server.

Aggregator: is a role that looks at data in the SACM environment, combines together pieces of related data and publishes the new relationships back into the environment. An example of what an aggregator might do is to look at the information provided by a DHCP server along with historic IP address information for a machine to determine when records referring to a machine are the same one or different ones.

Evaluator: is a role that looks at the data in the SACM environment along with a set of evaluation criteria, compares the data with the criteria and publishes the results of the evaluation back into the environment. Evaluators can look at things as simple as are all of the pieces of software on a machine licensed to as complicated as comparing data for a network or machine against an intrusion report.

SACM Proxy: is a role that allows for data to be transferred between two different SACM environments. SACM proxies frequently provide the data store role as well so that they can know what data needs to be synchronized between the two subsystems. SACM proxies can exist in environments where they are always active, or where only intermittent connectivity exists between the two subsystems.

Policy Management: is a role that controls different policy configurations for the environment. This includes such things as: What to report and who to report it to. Conditions under which automated remediation should automatically be applied. What the priorities are used in performing evaluations and frequencies of evaluations.

External Event: is a role that acts between the SACM environment and external non-SACM systems. One type of interaction that would be covered here is dealing with external vulnerability reports. As reports come from external sources, they need to be messaged into the correct format for storage in the SACM IM and then supplied to evaluators to identify if the vulnerability exists in systems SACM monitors. The same thing can happen in reverse, where vulnerabilities or conditions that look like they might be

vulnerabilities could be reported to an external centralized authority.

In many cases it is not always clear where a single service falls. A simple example of this is where the NEA protocol would fit into the picture. NEA can be treated either as an internal collector or as an external collector depending on what is being collected and how the observer feels about the world. Many of the core things that NEA collects today are not directly mapped to the SACM Information Model without some massaging, as such it is reasonable to look at this as being an external collector with the NEA server being co-resident with the SACM External Collector. This entity would take the NEA data, convert it into the SACM IM, and then publish it using one of the SACM DMs. On the other hand, it is also possible that NEA would collect data that corresponds directly to a SACM IM and return the data already encoded in a SACM DM. In this case, it would make sense to consider the NEA client as being a SACM Internal collector which publishes using NEA as the publishing protocol and the ENA server being either a proxy or a data store.

[2.1.](#) Combining Roles

In many cases more than one role will be combined into a single entity. In this section a view of how roles might be combined together to provide support as a proxy for the Ice Station Zebra use case in [[RFC7632](#)]. In this scenario, the enterprise has two different SACM sub-systems. One sub-system is at the main enterprise and the other sub-system is at Zebra. There is not a constant high-speed connection between the two sub-systems, instead data is exchanged an intermittent, low-speed, high-latency link.

The proxy role is the one that sits between the two subsystems, that is what it is designed for. Due to the link, one would have two different proxies one on either side of the link. The entity that contains the proxy may additionally have the following roles:

- o Data Store: Placing a data store on the proxy allows for queries about devices on the other side to be queued up and synchronized across the connection. Additionally, having a data store for those items on the opposite side of the link allows that data

store to specialize on how to optimally synchronize data across

the link.

- o Directory Service: Placing a directory service that tells about the systems and services on opposite side of the connection allows for the ability to filter down the set of registrations that are passed over the link.
- o Authorization Service: The authorizations may be changed for some entities based on the location of the service they are asking about. A user may have basically unlimited authorization for the local subsystem, but may have limited authorization for the remote subsystem. This rule would apply whether they were at the home base or at Zebra.

[2.2.](#) SACM Collection

The collection of data in the SACM world is one of the major issues that needs to be dealt with. For this reason we drill down into that problem here.

When dealing with just the issue of collection, the SACM architecture can be thought of as having a left-hand side and a right-hand side, illustrated by the dotted line in Figure 2. The left-hand side describes how management protocols can be used to allow a Collection Controller (CC) to choreograph the collection of endpoint posture from a set of target endpoints through posture change subscriptions or direct requests, and for endpoints to report posture information based on these collection requests. The left-hand side of the architecture is built upon existing endpoint management protocols; however, new protocols are needed to separate out the evaluation and collection capabilities defined by some of these protocols (e.g., NEA).

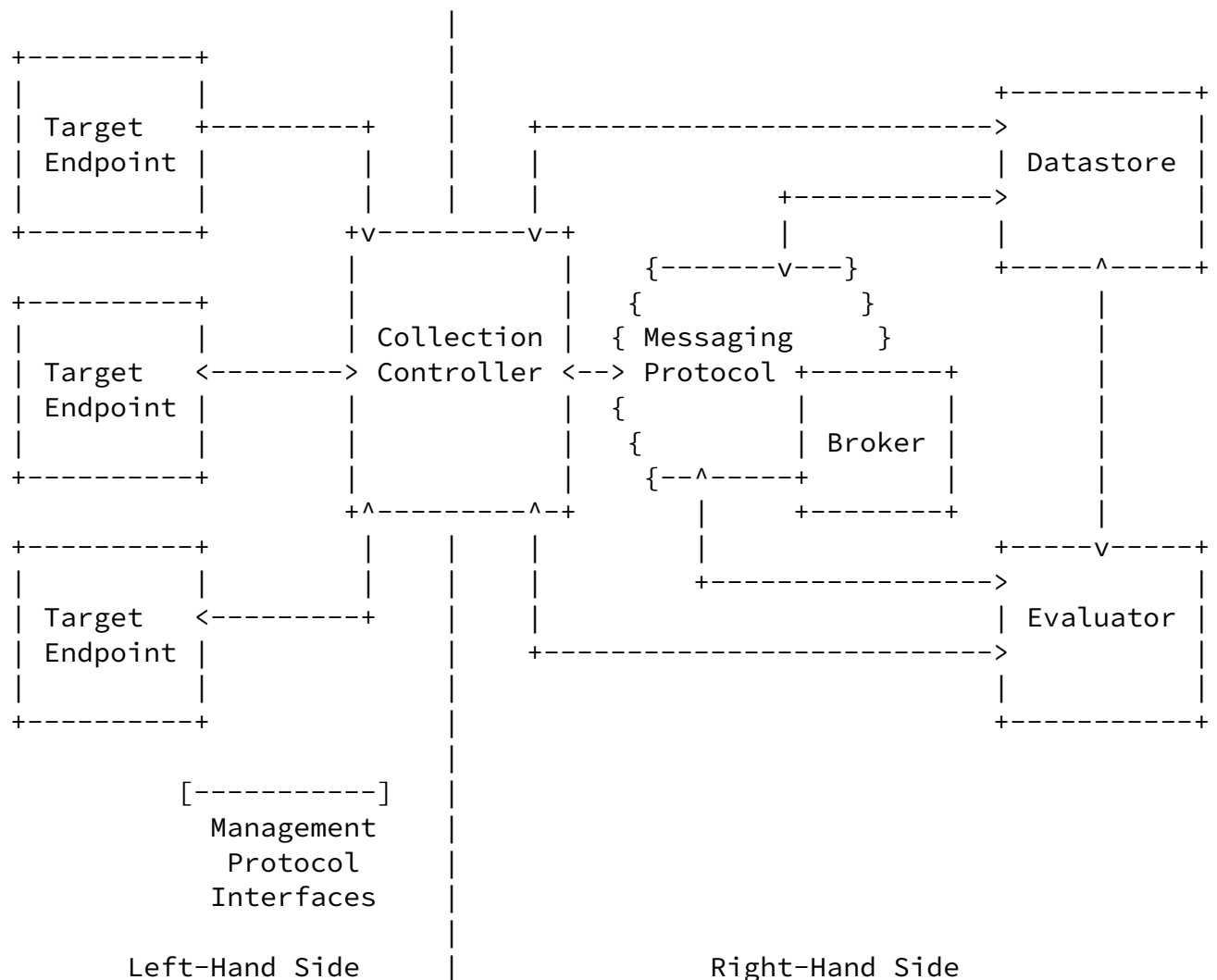


Figure 2: SACM Architecture

At the center of the SACM architecture is a CC that spans the left- and right-hand sides. The CC choreographs the collection of posture information on the left-hand side, and provides a common view of the collected posture for the right-hand side. Working in this way, all Evaluators that consume posture information from a given CC are provided a common view of all endpoint posture information collected by the CC. This common view can be accessed by Evaluators who need this information for asset management, configuration management, and vulnerability management use cases, and collected posture information can also be made available to other posture consumers to support unforeseen use cases.

The right-hand side of the SACM architecture defines how posture information is shared between components that provide collected posture (e.g., a Collection Controller) and components that store,

evaluate, and use collected posture information on the network. Analytical and reporting capabilities need to be able to publish and

subscribe to posture data feeds from other tools that have the required information. Other capabilities may need to access data stores of previously collected, historic posture information. By separating the methods of posture collection from how this information is consumed for use, the SACM architecture protects endpoints from being overloaded by downstream requests, and posture data from unauthorized access. These benefits support security automation by providing improved access to posture information by downstream consumers, and allowing downstream components to communicate information derived from posture analysis.

The following subsection discuss the motivations for the SACM architecture.

2.2.1. Use of Existing Management Protocols

On the left-hand side of the SACM architecture, a number of existing, widely deployed endpoint management protocols support the collection of posture information from endpoints. Examples of these protocols include: the Simple Network Management Protocol (SNMP) [[RFC3411](#)], Network Configuration Protocol (NETCONF) [[RFC4741](#)], and the Network Endpoint Assessment (NEA) protocol [[RFC5209](#)]. These and similar protocols provide extension points that allow for new types of posture information to be represented and collected over time. For these reasons, integration with existing posture collection protocols is a key feature of this architecture.

A gap in the functionality provided by existing protocols is a generalized mechanism to allow external components to drive data collection activities through a common, protocol agnostic collection interface. This is a feature supported by the SACM architecture through the definition of a common interface on the right-hand side that allows the CC to choreograph posture collection through implementations of existing management protocols on the left-hand side.

2.2.2. The Need for Collection Choreography

Collection choreography is the act of managing posture collection

activities on the left-hand side to produce a common view of collected posture information for one or more collection consumers on the right-hand side. Collection choreography is supported in the SACM architecture by the Collection Controller (CC) for the following reasons:

Simplification of Collection Clients Multiple management protocols are needed to fulfill posture information collection requests across different types of endpoints. To provide coverage of

all posture information that may need to be collected, any component driving posture collection must be well versed in multiple protocols and associated data models. Developing and maintaining this level of protocol support is a heavy burden to place on Evaluators. Additionally, some management solutions make use of specialized collection managers or proxies (e.g., mobile device management) that directly communicate with the target endpoint(s) on behalf of requesting clients. Requiring evaluators to locate and communicate with these proxies would discourage development and adoption of SACM solutions. The SACM architecture reduces the implementation burden on Evaluators by making the CC manage which collection services a request should be sent to when satisfying a given SACM collection request. This allows Evaluators to focus only on identifying and processing the data they request, and not on managing the collection process.

Authenticating, Authorizing, and Controlling Access of Data Collection Consumers

An automated collection system can pose a significant risk to an organization if the system can be misused by an attacker to gain knowledge of the operational state of endpoints. Once accessed by an attacker, this information can be used to attack other hosts or move laterally within a network. For this reason, access to collected endpoint posture information must be restricted to authenticated and authorized entities. Different management protocols may employ different authentication and authorization schemes. Through the use of a shared CC, access can be controlled in a way that unifies the underlying authentication, authorization, and access control schemes.

Reducing Performance Impacts on the Target Endpoint Without a common CC, Evaluators would need to directly interact with an endpoint to request posture data. Multiple clients operating in this way might request the same posture data within a narrow time window, resulting in multiple collection operations that can reduce the operational performance of the endpoint. Use of a CC can allow these repetitive and duplicative collection operations to be optimized. Working in this way, the CC can compute an optimal collection approach, using the appropriate management protocols to efficiently collect any needed posture data. This data can be cached by the CC, operating as a Data Store, and can be reused within an appropriate time window to provide multiple clients with the same collected posture information.

For these reasons a Collection Controller is needed to choreograph posture collection within the SACM architecture.

[2.2.3.](#) Collected Information Dissemination

On the right-hand side of the SACM architecture, once collection has been triggered and choreographed, there remains the issue of efficiently disseminating the collected posture information to consumers. The collection activity will have yielded information in an arbitrary data format, often spread across several formats with overlapping and mismatched information. There is a need for a means to package this information in a standardized way such that automated systems downstream can consume it without a priori knowledge of the source format.

The SACM architecture needs to support direct request response and publish subscribe mechanisms for posture data dissemination. In either case, arbitrary data formats can be processed at collection time and re-enveloped by a standardized information description. The downstream consumer needs to only understand the enveloping model to process the message containing posture information. It might also be possible to allow the enveloping model to identify multiple formats, allowing the consumer to request information from the designated source in a specific format it recognizes and supports.

[3.](#) SACM Protocols

List of protocols:

Authentication: A method of doing an entity to authenticate itself is required. Several potential candidates exist for this purpose. Among these candidates are X.509 certificates, SAML assertions, EAP or GSS-API.

Authorization: A standardized method of either querying the authorization server or carrying authorization information needs to be selected. Some of the candidates for this would be SAML assertions or X.509 attribute certificates.

Query: There will exist a standardized method for querying data from a repository. The query protocol needs to do the following things:

Get the required authentication for the requester. The server needs to authenticate both that the requester has the needed rights both to make the query and to have access to the data involved.

The entities need to negotiate a data model to be used for the creation of the query and to express the query in.

The entities need to negotiate a data model, a serialization abstraction and a serialization data format for data to be returned in response to a query.

The ability to specify required meta-data as part of the query is a requirement. As an example, the ability to require that the data be collected after a given date and time is required.

The ability to negotiate the frequency that the data is to be returned to the client. Data may be required as a one time event or as an ongoing event. One time events may be returned either immediately, or when the query can be completed as the data becomes available. Ongoing responses can be based on a time interval or based on some event happening. For ongoing response, the criteria for specify the frequency of response

needs to be both negotiated and authorized.

Response: There will exist a standardized method for sending data from a server to a client. Data may be sent to a client based either on a request for data, a prior request for data or because the server decided the client needs the data. Servers need to do the authentication and authorization on the data to be returned at a regular basis.

Directory Register: There will exist a standardized method for registering and unregistering data in a directory service.

Directory Query: There will exist a standardized method for querying the directory service for supported services.

Publish: There will exist a standardized method for publishing data into a repository. The method will:

- * Verify the identity and authorization of the publisher.
- * If needed, negotiate the format in which the data is presented. This includes the data model, data abstraction and data serialization.
- * Verify that the required meta-data that the publisher must provide is present.
- * Provide repository based meta-data.

Additionally, the repository will potentially trigger events within the server for dealing with outstanding requests for data or evaluations.

Synchronization: There will exist a standardized method for doing synchronization between multiple data stores. This method will:

- * TBD.

Collection: Many different protocols can be used to do collection over. One of these is NEA [[RFC5209](#)] which acts as a transport for

serializations of SACM data models. The use of NEA as the transport satisfies the following requirements:

- * NEA requires the use of TLS [TLS??] for cryptographic protection between the client and the server.
- * NEA has defined two authentication protocols to run between the client and the server. These are PT-TLS [[RFC6876](#)] and PT-EAP [[RFC7171](#)]. PT-TLS uses X.509 certificates on both the client and server side to authenticate each side. PT-EAP allows for EAP [[RFC3748](#)] to be used which allows for lighter weight registration and authentication to occur.
- * Authorization is currently of the all or none variety. Authorization is done at configuration time.
- * Discovery is not currently supported by NEA, so the server location is configured onto the client.

[I-D.coffin-sacm-nea-swid-patnc] represents one set of attributes to carry SACM information.

[4.](#) Information Model

The SACM architecture includes a single Information Model to provide a consistent view of the data needed to do perform the endpoint posture assessment. Additional IMs may be defined by entities other than the IETF which are supersets of the IETF IM. This is one way that companies will be able to differentiate their products from each other. The IM includes not only the data itself, but metadata as well. The meta data includes, but is not limited to:

- o Who collected or created the data
- o When the data was collected or created
- o What data was used to create the data

- o What relationships exist between different data points or different versions of the same data point.

[5.](#) Data Model

The SACM architecture includes a single Data Model defined by the IETF. Additional DMs maybe defined by entities other than the IETF. These additional DMs may be either subsets or supersets of the IETF SACM DM. The subset DMs are defined for doing special purpose work which does not need the full SACM DM.

The term DM is used by the SACM group in two different contexts and it is useful to discuss those two contexts. Data Model can refer to how the data is arranged within a entity. In this view of a DM, one can use a relational database as the conceptual view of the world. The relations in the database reflect the relations between the data in the IM. The use of a relational database is not the only way to implement the DM within a process.

The term DM can also refer to how the data is represented when serialized for transport between two entities. This is the meaning that is more usual when looking at the current SACM documents. This is sometimes referred to as the DM in transit.

6. IANA Considerations

This document has no IANA Considerations.

7. Security Considerations

This document is a high level description of the architecture of the SACM environment, as such it does not directly specify any security problems or requirements. Security requirements for SACM can be found in other documents including the SACM Requirements [[I-D.ietf-sacm-requirements](#)].

8. Informational References

- [I-D.ietf-sacm-requirements]
Cam-Winget, N. and L. Lorenzin, "Security Automation and Continuous Monitoring (SACM) Requirements", [draft-ietf-sacm-requirements-14](#) (work in progress), September 2016.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", [RFC 3444](#), DOI 10.17487/RFC3444, January 2003, <<http://www.rfc-editor.org/info/rfc3444>>.

- [RFC7632] Waltermire, D. and D. Harrington, "Endpoint Security Posture Assessment: Enterprise Use Cases", [RFC 7632](#), DOI 10.17487/RFC7632, September 2015, <<http://www.rfc-editor.org/info/rfc7632>>.
- [RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", [RFC 5209](#), DOI 10.17487/RFC5209, June 2008, <<http://www.rfc-editor.org/info/rfc5209>>.
- [RFC6876] Sangster, P., Cam-Winget, N., and J. Salowey, "A Posture Transport Protocol over TLS (PT-TLS)", [RFC 6876](#), DOI 10.17487/RFC6876, February 2013, <<http://www.rfc-editor.org/info/rfc6876>>.
- [RFC7171] Cam-Winget, N. and P. Sangster, "PT-EAP: Posture Transport (PT) Protocol for Extensible Authentication Protocol (EAP) Tunnel Methods", [RFC 7171](#), DOI 10.17487/RFC7171, May 2014, <<http://www.rfc-editor.org/info/rfc7171>>.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", [RFC 3748](#), DOI 10.17487/RFC3748, June 2004, <<http://www.rfc-editor.org/info/rfc3748>>.
- [I-D.coffin-sacm-nea-swid-patnc]
Coffin, C., Haynes, D., Schmidt, C., and J. Fitzgerald-McKay, "Software Inventory Message and Attributes (SWIMA) for PA-TNC", [draft-coffin-sacm-nea-swid-patnc-03](#) (work in progress), October 2016.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, [RFC 3411](#), DOI 10.17487/RFC3411, December 2002, <<http://www.rfc-editor.org/info/rfc3411>>.
- [RFC4741] Enns, R., Ed., "NETCONF Configuration Protocol", [RFC 4741](#), DOI 10.17487/RFC4741, December 2006, <<http://www.rfc-editor.org/info/rfc4741>>.

Acknowledgments

This document is currently the opinions of just the authors and is not representative of the SACM working group or the IETF.

Internet-Draft

Prospective Architecture for SACM

October 2016

This document has been formed by discussions with a large number of the SACM working group members including but not limited to: Henk Birkholz, Lisa Lorenzin, Nancy Cam-Winget.

Authors' Addresses

Jim Schaad
August Cellars

Email: ietf@augustcellars.com

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Email: david.waltermire@nist.gov

Schaad & Waltermire

Expires May 4, 2017

[Page 17]